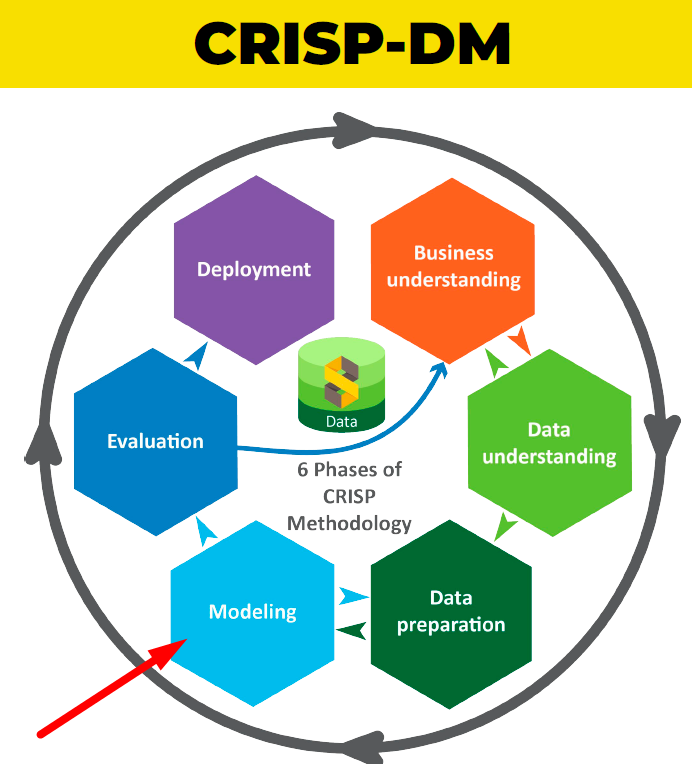
**Quarta fase do Crisp-dm Definindo Modeling**.

O modeling é a fase aonde vamos efetivamente **construir** o modelo, **selecionar** o modelo e **calibrar** os parâmetros dele (hyperparameters). **Avaliar** o modelo. Visão técnica.

**O que é o hyperparametros.**

É um termo mais detalhado dos parâmetros que controlam o aprendizado do modelo.

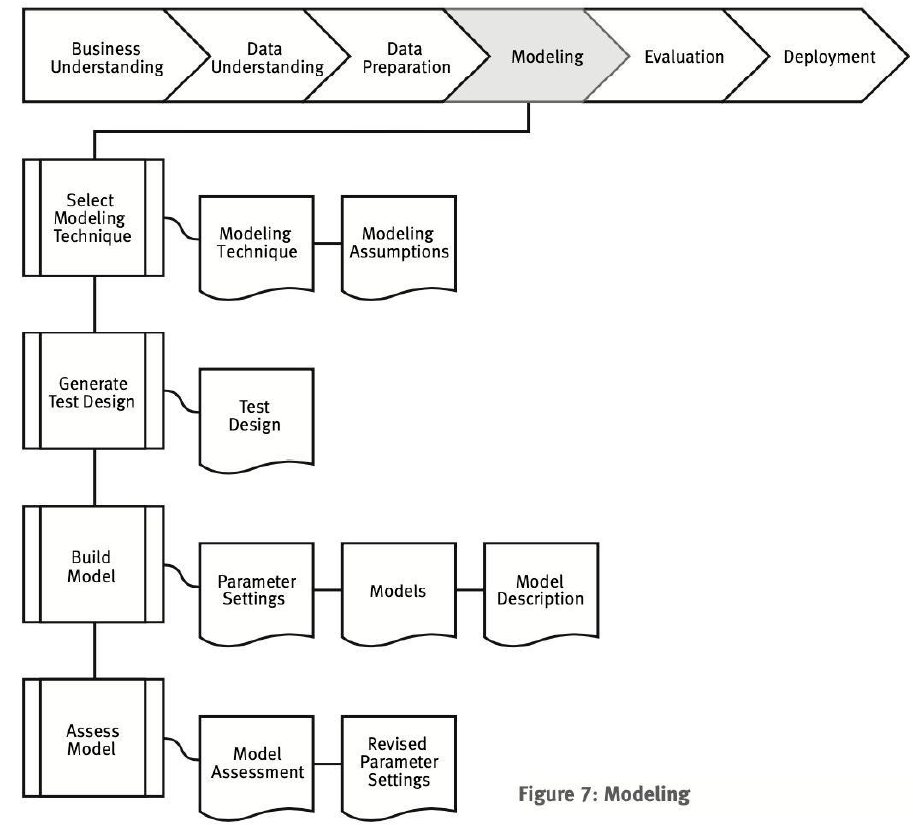
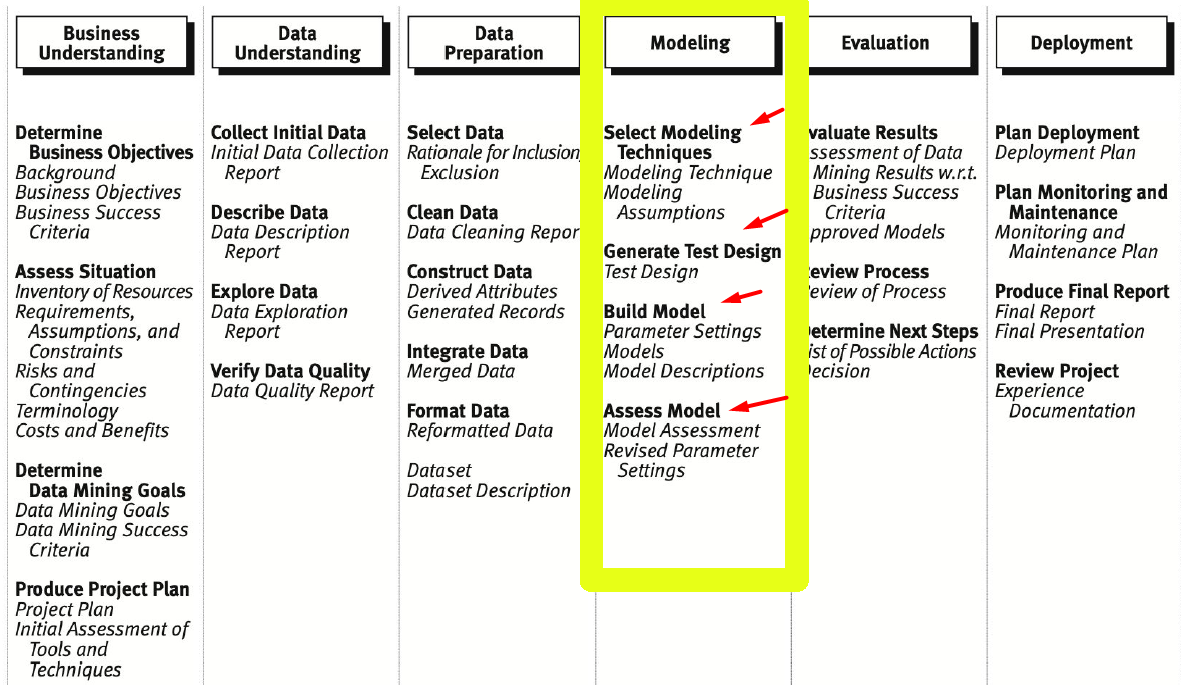
Parâmetros da função que está sendo executada.

Esse é um termo muito amplo. Nesse caso está falando de um parâmetro especifico que controla função dentro do machinne learning.



São os parâmetros que vai na função que definimos e ajustaremos e trabalhar com eles.

**Selection Modeling Technique**

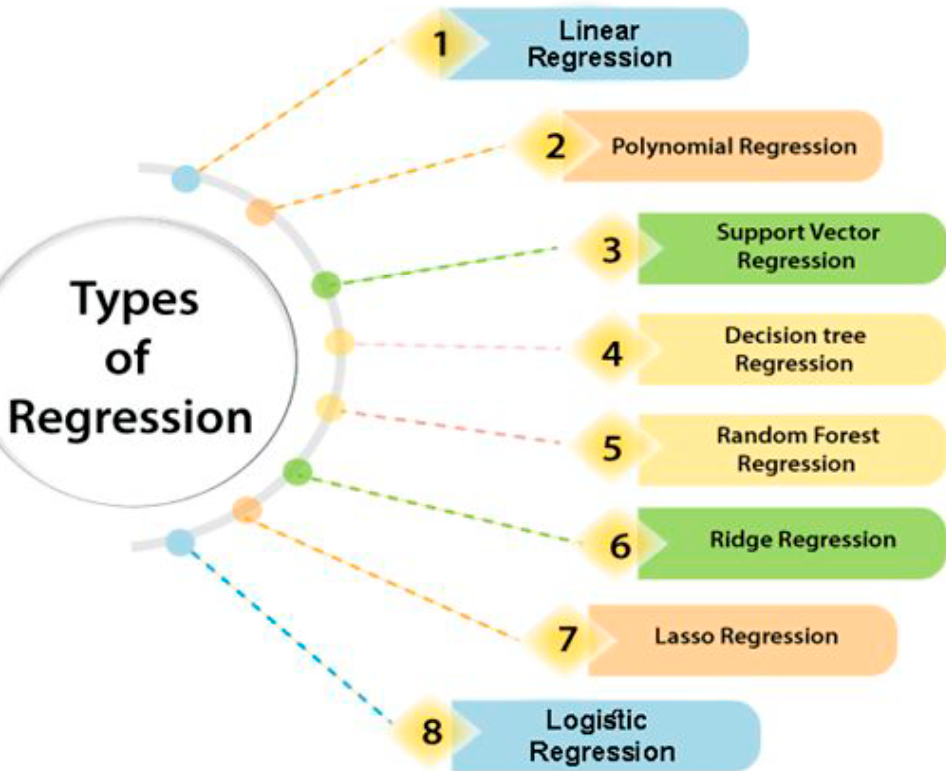


Essa task possui 2 atividades, o **Modeling techniche** e o **Modeling assumptions**.

Será onde vamos realizar a seleção da técnica do modelo. Refere-se a uma técnica de modelagem especifica.

Não é uma classificação e possui várias para o nosso tipo de problema.

Se forem várias técnicas que executaremos e aplicar, execute estas tarefas separadamente para cada técnica.



Temos que saber bem qual modelo usaremos e de preferência, qual biblioteca que também usaremos.

**Output**

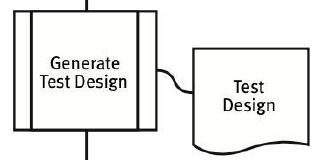
**Modeling technique**, documentar qual foi a técnica que foi selecionada. É importante para o nosso time e para min também.

**Modeling Assumptions**: Premissas da técnica selecionada. Ex: Não aceita valores categóricos. Descrever essas premissas técnicas do modelo que selecionamos.

Essa fase de selecionar técnicas é uma caixa de ferramentas com vários tipos de modelos, onde devemos ler e pesquisar um pouco sobre como e o que usaremos no nosso modelo.

Não é bom selecionar técnica muito distintas porque será difícil de avaliar qual técnica funcionou melhor.

**Generate Test Design**

****

Gerar o design de teste, e nessa task possui uma atividade de mesmo nome, a Test design.

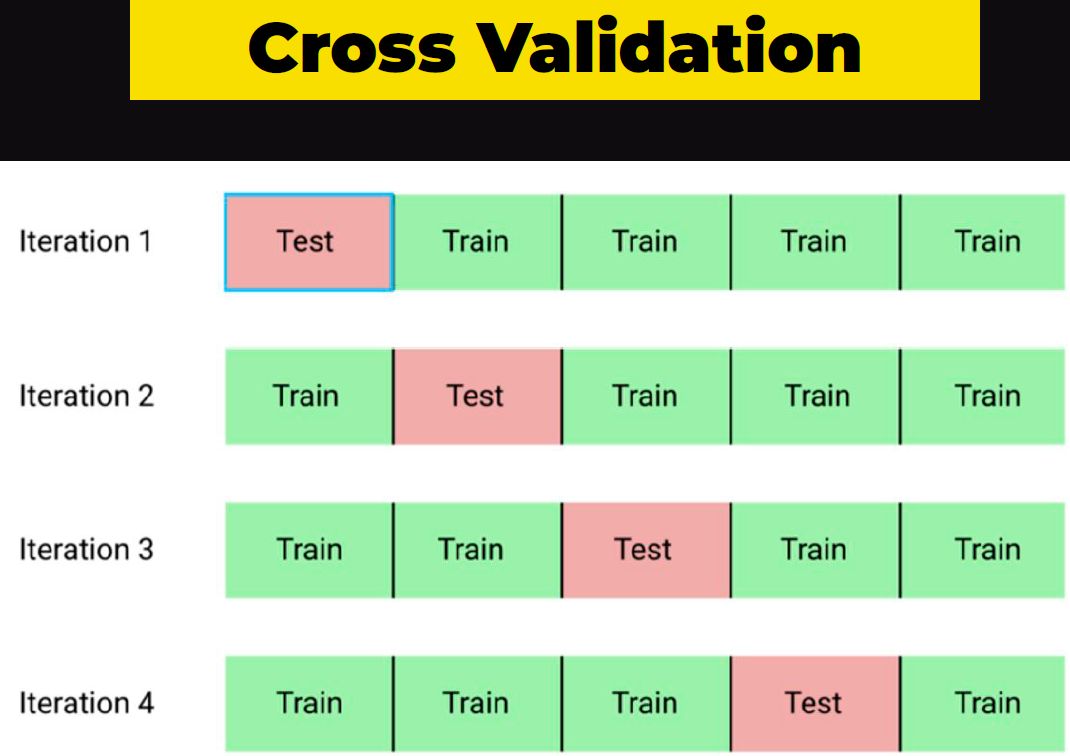
Nessa task construirei o teste do modelo, onde definiremos como que avaliaremos esse modelo e é um teste intrínseco o interno próprio, não é um modelo que irá funcionar na vida real é específico para teste.

E será importante analisar R square ou F1-Score e também como irá separar os dados para o modelo.

E como que será feito isso iremos separa o teste dataset.

Ex:

Ou um cross validation:



Depois veremos quais métricas que vamos avaliar se será um modelo de F1-Score onde quero penalizar o modelo quando ele tem um falso positivo ou um falso negativo.

E vai definir qual dato será separado para esses testes.

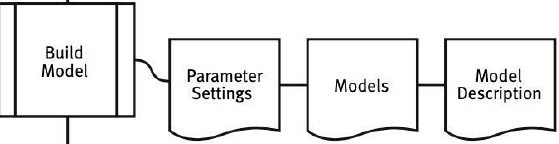
**Output Test design**:

Justamente descrever qual o plano de treinamento, o teste e a validação que será implementada.

E um ponto importante do output é determinar como o dataset será dividido em train, test e validation set.

O crisp não fala qual técnica que vamos utilizar, só diz o que é necessário de ser feito.

**Build Model**

****

Construção do modelo em si, e é uma task curta com 3 outputs (parameter settings, models, model description).

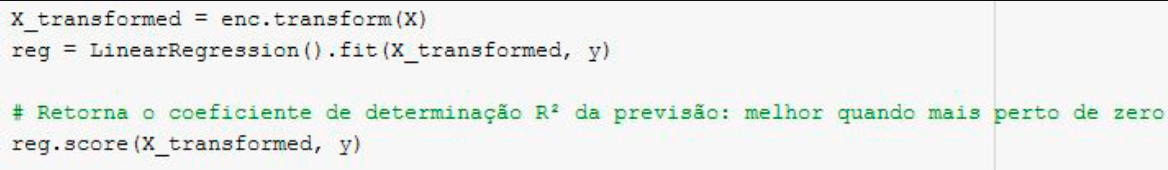
Essa task é efetivamente executar as ferramentas de modelagem sobre o teu dataset preparado, muitas vezes o build model é feito em um, dois ou três linhas.

Pode também utilizar modelos prontos.

Ex:

Primeira linha estamos criando uma variável nova chama X-transformed que é o X com uma transformação em cima que está sendo aplicada aqui e na sequência temos regressão é igual a linear regression . fit (x-transformed e o Y) basicamente tudo em uma linha de código.E a variável REG é justamente o nosso modelo.

No fim tem o regression ponto score para fazer o teste a avaliação do nosso modelo.



Temos 3 outputs.

**Parameters settings**

É configuração dos parâmetros de cada fórmula de modelos, ela tem alguns hyparamentros a ser definido.

Esse output lista com os parâmetros definidos e o valor deles.

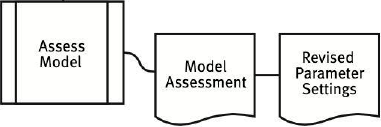


**Modelo & model description.**

O Modelo é próprio modelo é o output, o exemplo é o REG.

Model description que é uma descrição desse modelo, utilização e documentação.

**Assess Model**



Fazemos a avaliação do modelo. De acordo com os critérios de sucesso da mineração de dados, e de acordo com o design de testes.

Essa fase é a avaliação apenas do modelo. No evaluation tem uma avaliação do todo.

Avaliação das técnicas diferentes e dos modelos, vendo os modelos que vamos seguir lá no evaluation.

É uma parte apenas de avaliação do modelo.

O outputs são:

**Model assessment**

Sumarização com o resultado da análise, documentando pode ser apenas em uma linha ou criando uma célula de texto no código mesmo. Documentação no meu código e mostrando qual técnica utilizamos.

**Revised parameter settings**

Revisar os hyperparametros do modelo e realizar ajustes para buildar novamente o modelo. Vai olha o que pode melhorar ou se vai melhorar fazendo testes e verificando se funcionou no meu modelo para ter um modelo mais assertivo.

Essa iteração deve ocorrer até encontrar o melhor modelo.

O Crisp -DM não nos dá o passo a passo.

Exemplo prático: <https://drive.google.com/file/d/1SZ5IuXbjqZgk4cWNzwm4NBZtKOzIxHN6/view?usp=sharing>

Resumo:

**Modeling** aonde vamos efetivamente construir o nosso modelo.

E o primeiro passo é **selecionar o nosso modelo**.

**Generate test design** que é onde vamos definir como vamos separar o nosso dataset (dataset de treino, dataset de teste) vai fazer isso via **K-fold cross validation** e definir qual é a melhor técnica.

**Build model**, é a parte mais simples em no máximo duas linhas de código. Depois fazendo o ponto fit e pode fazer tudo em uma linha só. Tem o modelo e treinou o modelo e vemos as infinidades de hyperparametros para

analisar e depois vimos o otimizador de parâmetros.

**Assess model** depois de escolher o melhor modelo, para descrever, as métricas e depois otimizar os parâmetros desse modelo.

É um caminho para seguir o que é importante para o nosso projeto.

**AutoML**